Table of Contents

Compilers

Intel Compiler

DRAFT

This article is being reviewed for completeness and technical accuracy.

Intel compilers are recommended for building your applications on either Pleiades or Columbia.

On Columbia, a system default version has been loaded automatically. On Pleiades, there is no system default--you must load a specific module. Use the "module avail" command on Pleiades to see what versions are available and load an Intel compiler module before compiling. For example:

```
% module load comp-intel/11.1.072
```

Notice that when a compiler module is loaded, some environment variables, such as FPATH, INCLUDE, LD_LIBRARY_PATH, etc., are set or modified to add the paths to certain commands, include files, or libraries, to your environment. This helps to simplify the way you do your work.

To check what environmant variables will be modified for a module, do, for example:

```
% module show comp-intel/11.1.072
```

On Columbia and Pleiades, there are Intel compilers for both Fortran and C/C++:

- **Intel Fortran Compiler: ifort (version 8 and above)**

  The ifort command invokes the Intel(R) Fortran Compiler to preprocess, compile, assemble, and link Fortran programs.

  ```
  % ifort [options] file1 [file2 ...]
  ```

  Read **man ifort** for all available compiler options.

  To see the compiler options by categories, do:

  ```
  % ifort -help
  ```

file*N* is a Fortran source (.f .for .ftn .f90 .fpp .F .FOR .F90 .i .i90), assembly (.s .S), object (.o), static library (.a), or other linkable file.

Source Files Suffix Interpretation:

- .f, .for, or .ftn : fixed-form source files
- .f90 : free-form F95/F90 source files
- .fpp, .F, .FOR, .FTN, or .FPP: fixed-form source files which must be preprocessed by the fpp preprocessor before being compiled
- .F90 : free-form Fortran source files which must be pre-pro- cessed by the fpp preprocessor before being compiled

- **Intel C/C++ compiler: icc and icpc (version 8 and above)**

The Intel(R) C++ Compiler is designed to process C and C++ programs on Intel-architecture-based systems. You can preprocess, compile, assemble, and link these programs.

```
% icc [options] file1 [file2 ...]
% icpc [options] file1 [file2 ...]
```

Read **man icc** for all available compiler options.

To see the compiler options by categories, do:

```
% icc -help
```

The icpc command uses the same compiler options as the icc command. Invoking the compiler using icpc compiles .c, and .i files as C++. Invoking the compiler using icc compiles .c and .i files as C. Using icpc always links in C++ libraries. Using icc only links in C++ libraries if C++ source is provided on the command line.

file*N* represents a C/C++ source (.C .c .cc .cp .cpp .cxx .c++ .i), assembly (.s), object (.o), static library (.a), or other linkable file.

GNU Compiler Collection

DRAFT

This article is being reviewed for completeness and technical accuracy.

GCC stands for "GNU Compiler Collection". GCC is an integrated distribution of compilers for several major programming languages. These languages currently include C, C++, Objective-C, Objective-C++, Java, Fortran, and Ada.

The GNU C and C++ compiler (gcc and g++) and Fortran compiler (gfortran) through the Linux OS distribution are available on Pleiades and Columbia. The current version installed (under /usr/bin) can be found with the following command:

```
% gcc -v
 ... gcc version 4.1.2 20070115 (SUSE Linux)
```

Newer versions of GNU compilers can be requested and installed as modules. Currently, there is a gcc/4.4.4 module, which includes gcc, g++, and gfortran, available on Pleiades.

Read **man gcc** and **man gfortran** for more information.